2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

Battery Swapping Assignment for Electric Vehicles: A Bipartite Matching Approach

Pengcheng You, John Z. F. Pang, Minghua Chen, Steven H. Low, and Youxian Sun

Abstract— **This paper formulates an optimal station assignment problem for electric vehicle (EV) battery swapping that takes into account both temporal and spatial couplings. The goal is to reduce the total EV cost and station congestion due to temporary shortage in supply of available batteries. We show that the problem is reducible to the minimum weight perfect bipartite matching problem. This leads to an efficient solution based on the Hungarian algorithm. Numerical results suggest that the proposed solution provides a significant improvement over a greedy heuristic that assigns nearest stations to EVs.**

I. Introduction

Transportation emits a large amount of greenhouse gases, e.g., about a quarter of all greenhouse gases in the US [1], [2]. Electrification will greatly reduce the carbon footprint of transportation especially with increasing renewable generation of electricity. EVs are large loads that can add significant stress to electricity grids, but they are also flexible loads that can help mitigate the volatility of renewable generation through smart charging. There has been a large literature on the optimization of EV charging, e.g. [3]–[10]. EV charging however takes a long time. It is not suitable for commercial vehicles, such as taxis, buses, and ride-sharing cars, that are on the road most of the time, the opposite of most private cars.

An alternative EV refueling method is battery swapping where an EV swaps its depleted battery for a fully-charged battery at a service station.[1] This can be done in a few minutes. Several such electric taxi programs are in pilot in China [11]. The literature on EV battery swapping is small. In [12] the operation of a battery charging and swapping station is modeled as a mixed queuing network, consisting of an interior closed queue of batteries going through charging and swapping, and an exterior open queue of EV arrivals. Using this model, [13] proposes an optimal charging policy. An optimal assignment problem is formulated in [14], [15] that assigns to a given set of EVs best stations to swap their batteries based on their current locations and states of charge. The assignment aims to minimize a weighted sum of total travel distance and generation cost over both station assignments and power flow variables, subject to EV range constraints, grid operational constraints and AC power flow equations. While [14], [15] focus on spatial optimization over power grid operation within a single time slot, this paper focuses instead on temporal optimization where EVs arrive over several time slots.

Specifically, we adopt a discrete time model [16]. In each time slot, a centralized operator optimally assigns stations to a set of EVs that need battery swapping. Consider the optimal station assignment problem at time slot 1 where stations are assigned in a way that minimizes both the total EVs' cost to travel to their assigned stations and the total congestion (battery shortage) levels at these stations. The current assignment at time slot 1 will determine the EV arrival processes, and hence the congestion levels, at the stations in the future and in turn needs to take into account congestion due to EV arrivals at these stations that were scheduled before time slot 1. The problem is a binary program with strong temporal and spatial couplings. We show that it is polynomial-time solvable by reducing it to the standard minimum weight perfect bipartite matching problem. This leads to a solution based on the Hungarian algorithm for bipartite matching problems. We present numerical results that demonstrate that the proposed solution provides a significant benefit over a greedy heuristic that assigns to each EV its nearest station.

The remainder of this paper is structured as follows. Section II formulates the optimal station assignment problem for EV battery swapping, followed by the proposed polynomial-time solution in Section III. Section IV validates the theoretical analysis via numerical results, while Section V concludes.

II. Problem Formulation

Consider a group of EVs, e.g., a fleet of electric taxis, with swappable batteries that swap their depleted batteries for fully-charged ones at stations assigned by a central operator. Time is slotted with a constant length, e.g., 10 minutes. Without loss of generality, fix the current time slot as time slot 1 of the time horizon $\mathbb{T} := \{-T_m+1, \dots, 0, 1, \dots, T_m\}$, and let $\mathbb{T}^+ := \{1, \dots, T_m\}$, which is the segment of $\mathbb{T}$ from the current time slot on. Correspondingly, $\mathbb{T}\backslash\mathbb{T}^+$ refers to past time slots. T_m is a constant which we will interpret later. Suppose there is a set $\mathbb{J} := \{1, \dots, J\}$ of stations that provide battery swapping service for EVs. Denote the current number of (fully-charged) batteries that are available for swapping as n_j^0 at station j.

At the current time slot 1, let $\mathbb{I} := \{1, \dots, I\}$ be the set of EVs that require battery swapping. Our goal is to optimally assign a station $j \in \mathbb{J}$ to each EV $i \in \mathbb{I}$,

P. You and Y. Sun are with the State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China pcyou@zju.edu.cn, yxsun@iipc.zju.edu.cn

J. Z. F. Pang, and S. H. Low are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA {jzpang, slow}@caltech.edu

M. Chen is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China minghua@ie.cuhk.edu.hk

[1]In this paper, stations refer to battery (swapping) stations.

such that a weighted sum of aggregate EV cost and station congestion is minimized. Note that if future information is somehow available, joint optimization of station assignments over multiple time slots can be easily accommodated.

A. Decision variables, states, and constraints

Let $M := (M_{ij}, i \in \mathbb{I}, j \in \mathbb{J})$ represent the station assignment to EVs,[2] where

$$M_{ij} = \begin{cases} 1 & \text{if station } j \text{ is assigned to EV } i \\ 0 & \text{otherwise} \end{cases}$$

We require that only one station be assigned to each EV, i.e.,

$$\begin{cases} \sum_{j\in\mathbb{J}} M_{ij} = 1, & i \in \mathbb{I} \\ M_{ij} \in \{0,1\}, & i \in \mathbb{I}, j \in \mathbb{J} \end{cases} \tag{1}$$

Note that we also use $M_{ij}(t)$, $t = -T_m+1, -T_m+2, \ldots, 0$, to represent past assignments, which are given. They affect the current assignment and are *parameters* in the problem formulation.

Let $\tau(\sigma, \delta, t)$ estimate the arrival time of an EV if it starts to travel at time slot t from origin σ to destination δ; thus $\tau(\sigma, \delta, t) \geq t$. It captures the time-dependent traffic conditions and in practice we largely care about $\tau(\sigma, \delta, 1)$ that captures the real-time traffic information. Meanwhile, $\tau(\sigma, \delta, 1)$ also corresponds to an optimal routing based on the current traffic conditions, thus we can readily obtain the associated travel distance from σ to δ, denoted by $d(\sigma, \delta, 1)$. Note that the current estimates $\tau(\sigma, \delta, 1)$ and $d(\sigma, \delta, 1)$ are available by resorting to, say, Google Maps. Their explicit modeling goes beyond the scope of this paper, thus we assume they are exogenous and given. On this basis, if EV i starts at time slot t, by defining its position as σ_i and the location of station j as δ_j, it is expected to travel the distance of $d_{ij}(t) := d(\sigma_i, \delta_j, t)$ and arrive at a future time slot $\tau_{ij}(t) := \tau(\sigma_i, \delta_j, t)$ at station j, thus reducing the available batteries at time slot $\tau_{ij}(t)$ by 1. We also define $\tau_{ij}^{-1}(t)$ as the inverse function of $\tau_{ij}(t)$, i.e., $\tau_{ij}^{-1}(t)$ is the time slot when station j was assigned to EV i that arrives at time slot t. For brevity, let $\tau_{ij} := \tau_{ij}(1)$ and $d_{ij} := d_{ij}(1)$. Note that τ_{ij} is also the travel time of EV i if it currently sets off to station j.

Now we interpret T_m as the maximum travel time of an EV to reach a station, i.e., $T_m := \max_{i,j,t}(\tau_{ij}(t)-t+1)$. T_m is tightly bounded as an EV that requires battery swapping is running out of energy and its maximum driving range is limited. The assignments before $-T_m + 1$ are summarized in n_j^0, and the states of stations after T_m will not be directly affected by the current assignment.[3]

Let $n_j(t)$ denote the number of available batteries at station j at the end of time slot t, which is the station state. In particular, $n_j(0) = n_j^0$. Hence $n_j(t)$ increases by 1 when a battery at station j becomes fully-charged, and decreases by 1 when a fully-charged battery is removed by an EV:

$$\begin{aligned} n_j(t) = n_j(t-1) + c_j(t) - \sum_{i\in\mathbb{I}_p} M_{ij}(\tau_{ij}^{-1}(t)) \\ - \sum_{i\in\mathbb{I}} M_{ij} \cdot 1(t = \tau_{ij}), \qquad t \in \mathbb{T}^+ \end{aligned} \tag{2}$$

where $c_j(t)$ is the number of batteries that become fully-charged at station j in time slot t (which is known a priori), and $\mathbb{I}_p$ is the set of all past EVs that were assigned stations during the time interval $[-T_m + 1, 0]$. $1(x)$ is an indicator function for the predicate x. The third and fourth terms on the right-hand-side of (2) summarize the impacts of past assignments and the current one, respectively, on the number of available batteries at station j at time slot t. The second and third terms are both given while the fourth one is to be decided. Moreover, past assignments have no effect on $n_j(T_m)$ by definition, i.e., $M_{ij}(\tau_{ij}^{-1}(T_m)) = 0$, $i \in \mathbb{I}_p$, $j \in \mathbb{J}$. Note that $n_j(t)$ can be negative. For instance, $n_j(t) = 5$ means that, at the end of time slot t, there will be 5 available batteries left after serving EVs that arrive at station j at time slot t; $n_j(t) = -3$ means there will be no available battery but 3 waiting EVs.

An EV can only be assigned a station within its driving range, i.e.,

$$d_{ij}M_{ij} \leq rs_i, \quad i \in \mathbb{I}, j \in \mathbb{J} \tag{3}$$

where r is the driving range per unit state of charge and s_i denotes the state of charge of EV i.

B. Optimal station assignment problem

The system cost has two components. First, a cost α_{ij} is incurred if station j is assigned to EV i, thus the cost of EV i is $\sum_{j\in\mathbb{J}} \alpha_{ij}M_{ij}$. For example, α_{ij} can be a weighted sum of EV i's travel distance and time from its current location to station j. Second, as explained above, $\langle -n_j(t)\rangle^+$ is the number of waiting EVs at the end of time slot t, where $\langle x\rangle^+ := \max\{x, 0\}$.

Let $n := (n_j(t), j \in \mathbb{J}, t \in \mathbb{T}^+)$ be the vector of station states. We are interested in the following *optimal station assignment* problem:

$$\begin{aligned} \min_{M,n} \quad & \sum_{i\in\mathbb{I}}\sum_{j\in\mathbb{J}} \alpha_{ij}M_{ij} + \sum_{j\in\mathbb{J}}\sum_{t\in\mathbb{T}^+} \langle -n_j(t)\rangle^+ \\ \text{s.t.} \quad & (1), (2), (3) \end{aligned} \tag{4}$$

which minimizes the weighted sum of aggregate EV cost and station congestion, subject to EVs' driving ranges.

III. Polynomial-time solution

The optimal station assignment problem (4) is a binary program with temporal couplings in (2) and spatial couplings implied in station congestion. It can however be solved efficiently.

***Theorem** 1:* The optimal station assignment problem (4) is polynomial-time solvable.

We prove the theorem in two steps. We first reformulate problem (4) in Section III-A into a more convenient mixed integer linear program (MILP), and then show in Section III-B that it is reducible to the problem of minimum weight perfect bipartite matching.

[2]More precisely, M_{ij} should be $M_{ij}(1)$ that denotes a current decision variable. We drop the notation of the current time slot for brevity.

[3]As shown later, they are affected through $n_j(T_m)$, which will be captured.

A. Reformulation as MILP

Note that all the constraints in (4) are linear in the variables (M, n). The only nonlinearity is $\langle -n_j(t)\rangle^+$, which can be removed by introducing an auxiliary variable $v_j(t)$ to replace aggregate station congestion by $\sum_j \sum_t v_j(t)$ and requiring $v_j(t)$ to satisfy the linear constraints $v_j(t) \geq 0$ and $v_j(t) \geq -n_j(t)$. Hence (4) is an MILP.

To reformulate it into a more convenient form, denote the number of available batteries at station j over $\mathbb{T}^+$ observed at time slot 1 before the current decision M is made by:

$$\tilde{n}_j(t) := n_j(0) + \sum_{\kappa=1}^{t}(c_j(\kappa) - \sum_{i\in\mathbb{I}_p} M_{ij}(\tau_{ij}^{-1}(\kappa))), \quad t \in \mathbb{T}^+$$

It is a known constant determined by past assignments. The evolution of $n_j(t)$ in (2) then reduces to

$$n_j(t) = \tilde{n}_j(t) - \sum_{i\in\mathbb{I}} M_{ij} \cdot 1(t \geq \tau_{ij}), \quad t \in \mathbb{T}^+ \tag{5}$$

which is decoupled across time slots, because $\tilde{n}_j(t)$ and the indicator function in (5) remove the dependency of $n_j(t)$ on $n_j(t-1)$.

The interpretation of $M_{ij} \cdot 1(t \geq \tau_{ij})$ in (5) is as follows. If station j is assigned to EV i at time slot 1, then it will arrive there at time slot τ_{ij}, thus removing one available battery from station j for time slot τ_{ij} and every time slot afterwards. For each station $j \in \mathbb{J}$, define an arrival matrix $A_j \in \{0,1\}^{T_m \times I}$ such that its (t, i) entry is

$$A_j(t,i) \quad := \quad 1(t \geq \tau_{ij})$$

For example, if the i^{th} column of A_j is $[0\ 0\ 1\ 1]^T$, it means EV i will arrive at station j at time slot $\tau_{ij} = 3$, thus removing an available battery for time slots $t = 3, 4$. Finally, let Π denote the set of M with $M_{ij} = 0$ if station j is outside EV i's driving range, i.e., $d_{ij} > rs_i$.

Putting the above together, (4) is then equivalent to the following MILP:

$$\begin{aligned} \min_{M\in\Pi, v\geq 0} \quad & \sum_{j\in\mathbb{J}}\sum_{i\in\mathbb{I}} \alpha_{ij} M_{ij} + \sum_{j\in\mathbb{J}}\sum_{t\in\mathbb{T}^+} v_j(t) \\ \text{s.t.} \quad & \sum_{j\in\mathbb{J}} M_{ij} = 1, \quad i \in \mathbb{I} \\ & v_j(t) \geq -\tilde{n}_j(t) + \sum_{i\in\mathbb{I}} A_j(t,i) M_{ij}, \quad j \in \mathbb{J}, t \in \mathbb{T}^+ \end{aligned} \tag{6}$$

To gain some intuition, consider the case where every station j faces heavy congestion, i.e. $\tilde{n}_j(t) \leq 0$. Then the inequality in the MILP will be tight in optimality. An optimal assignment is: for each $i \in \mathbb{I}$,

$$M^*_{ij} = \begin{cases} 1, & j = j^* := \arg\min_{j: d_{ij} \leq rs_i} \alpha_{ij} + \sum_{t\in\mathbb{T}^+} A_j(t,i) \\ 0, & j \in \mathbb{J}\backslash\{j^*\} \end{cases} \tag{7}$$

Note that j^* may not be unique, and ties are broken arbitrarily. Minimizing α_{ij} favors a nearest station, while minimizing $\sum_{t\in\mathbb{T}^+} A_j(t,i)$ favors a station with the latest time of arrival to avoid further congestion. Hence (7) strikes a compromise.

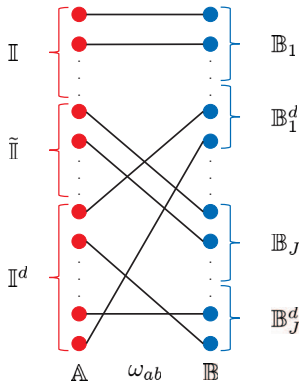

Fig. 1. An example of a perfect bipartite matching between the set of EVs and the set of batteries. The eventual assignment from the perfect bipartite matching is based on the subset of matches between real EVs and certain batteries. As the figure illustrates, dummy EVs can never have priority over real EVs for matching real batteries.

B. Reduction to Bipartite Matching

We now show that the MILP (6) can be further reduced to the minimum weight perfect bipartite matching problem, which is well known to be polynomial-time solvable. The intuition is that we let the EVs and individual batteries (not stations) be vertices on each bipartition, respectively, and assign a weight to each pair based on the contribution of the pair's matching to the objective value of (6).

Define a bipartite graph $\mathbb{G} = (\mathbb{A} \cup \mathbb{B}, \mathbb{E})$, where $\mathbb{A}$ and $\mathbb{B}$ are the bipartition of the vertex set and $\mathbb{E} \subseteq \mathbb{A} \times \mathbb{B}$ is the set of edges that are endowed with given weights $\omega := (\omega_{ab}, a \in \mathbb{A}, b \in \mathbb{B}, (a,b) \in \mathbb{E})$, as shown in Fig. 1. Without loss of generality, we assume $\mathbb{G}$ is *complete* and *balanced* as we can add infinite-weight edges and dummy vertices as necessary. Let $N := |\mathbb{A}| = |\mathbb{B}|$. The standard minimum weight perfect matching problem defined on $\mathbb{G}$ is

$$\begin{aligned} \min_x \quad & \sum_{(a,b)\in\mathbb{E}} \omega_{ab}\ x_{ab} \\ \text{s.t.} \quad & \sum_b x_{ab} = 1, \quad a \in \mathbb{A} \\ & \sum_a x_{ab} = 1, \quad b \in \mathbb{B} \\ & x_{ab} \in \{0,1\}, \quad a \in \mathbb{A}, b \in \mathbb{B} \end{aligned} \tag{8}$$

where $x := (x_{ab}, a \in \mathbb{A}, b \in \mathbb{B})$. Hence an instance of the bipartite matching problem (8) is specified by nodes $\mathbb{A}$, $\mathbb{B}$ and the weights ω.

Given an instance of the MILP (6), we now construct an instance of the bipartite matching problem (8) such that an optimal solution to (8) yields an optimal solution to (6).

- $\mathbb{A} := \mathbb{I} \cup \tilde{\mathbb{I}} \cup \mathbb{I}^d$. $\tilde{\mathbb{I}}$ is the set of EVs that were previously assigned stations, but have yet to have their batteries swapped (either on the way or waiting at stations). $\mathbb{I}^d$ is the set of dummy EVs if necessary to make $\mathbb{A}$ and $\mathbb{B}$ *balanced*.
- $\mathbb{B} := \bigcup_{j\in\mathbb{J}} \mathbb{B}_j \cup \mathbb{B}^d$. $\mathbb{B}_j$ is the set of available batteries at station j, including not only the currently available batteries, but also those that will become available in $\mathbb{T}^+$. The time slot when battery $b \in \mathbb{B}_j$ becomes

available is denoted as ρ_b, and $\rho_b = 0$ for the currently available batteries. $\mathbb{B}^d$ is the set of dummy batteries if necessary to make $\mathbb{A}$ and $\mathbb{B}$ *balanced*.

Remark 1: It is necessary to include $\tilde{\mathbb{I}}$, which comes from past assignments. Each EV $a \in \tilde{\mathbb{I}}$, while previously matched with a battery, currently only has a target station but no target battery. This is because battery matching is artificial for the purpose of an efficient solution to (6), and our original model still implies first-come, first-served. Consequently, EVs in $\mathbb{I}$ that arrive earlier will snatch batteries and may cause EVs in $\tilde{\mathbb{I}}$ to wait. This is captured in (6). To maintain consistency, we include $\tilde{\mathbb{I}}$ in $\mathbb{A}$, but restrict these EVs' matchings only with batteries at their originally assigned stations.

Remark 2: $\mathbb{I}^d$ and $\mathbb{B}^d$ are introduced to balance the bipartite graph. Note that $n_j(t)$ can be negative in (6). We have to make up the shortfall when $|\mathbb{I} \cup \tilde{\mathbb{I}}| > |\mathbb{B}_j|$ for station j by adding dummy batteries. More precisely, $\mathbb{B}^d := \bigcup_{j\in\mathbb{J}} \mathbb{B}^d_j$, where $|\mathbb{B}^d_j| = \max\{|\mathbb{I} \cup \tilde{\mathbb{I}}| - |\mathbb{B}_j|, 0\}$. Then $\mathbb{I}^d$ with $|\mathbb{I}^d| = |\bigcup_{j\in\mathbb{J}}(\mathbb{B}_j \cup \mathbb{B}^d_j)| - |\mathbb{I} \cup \tilde{\mathbb{I}}|$ is added to maintain balance between $\mathbb{A}$ and $\mathbb{B}$.

The nonnegative weight ω_{ab} of the match (a, b) corresponds to the incremental cost added to the objective of (6) if station j which battery b belongs to is assigned to EV a. To determine ω_{ab}, the main idea is to translate the congestion of stations to the waiting time each EV suffers.

1) $a \in \mathbb{I}, b \in \mathbb{B}_j$. Set $\omega_{ab} := \alpha_{ab} + \max\{\rho_b - \tau_{ab}, 0\}$.[4] Here, α_{ab} is the cost of EV a and $\max\{\rho_b - \tau_{ab}, 0\}$ is the time length for which it has to wait until battery b becomes available. If $d_{ab} > rs_a$, then $\omega_{ab} := \infty$.
2) $a \in \mathbb{I}, b \in \mathbb{B}^d_j$. Dummy batteries exist when there is a deficit in batteries to match all real EVs perfectly at station j. EVs matched with dummy batteries will wait until the end of $\mathbb{T}^+$ after their arrivals. Hence $\omega_{ab} := \alpha_{ab} + (T_m + 1 - \tau_{ab})$. If $d_{ab} > rs_a$, then $\omega_{ab} := \infty$.
3) $a \in \tilde{\mathbb{I}}, b \in \mathbb{B}_j$. We require that EVs $a \in \tilde{\mathbb{I}}$ stick to their original stations. To this end, if station j is originally assigned to EV a, $\omega_{ab} := \max\{\rho_b - \tau_{ab}, 0\}$; otherwise, $\omega_{ab} := \infty$. The EV cost is excluded since it does not contribute to the objective of (6).
4) $a \in \tilde{\mathbb{I}}, b \in \mathbb{B}^d_j$. Likewise, EVs $a \in \tilde{\mathbb{I}}$ are required to match dummy batteries at their original stations. Hence if station j is originally assigned to EV a, $\omega_{ab} := T_m + 1 - \tau_{ab}$; otherwise, $\omega_{ab} := \infty$.
5) $a \in \mathbb{I}^d, b \in \mathbb{B}_j$. Dummy EVs do not really exist, and should have no impact on the match result. Thus we have $\omega_{ab} := 0$.
6) $a \in \mathbb{I}^d, b \in \mathbb{B}^d_j$. Likewise, $\omega_{ab} := 0$.

From above, the parameters of (8) including N and $(\omega_{ab}, a \in \mathbb{A}, b \in \mathbb{B})$ can be computed in time of $O(N^2)$ given an instance of (6). On the other hand, if we have an optimal matching x^* for (8), an optimal assignment is straightforward:

$$M^*_{ij} = \sum_{b\in\mathbb{B}_j\cup\mathbb{B}^d_j} x^*_{ib}, \quad i \in \mathbb{I}, j \in \mathbb{J} \tag{9}$$

which is obtainable in time of $O(N)$.

Hence the optimal station assignment problem (4) is reduced to the minimum weight perfect bipartite matching problem (8). This proves Theorem 1.

C. Hungarian algorithm

The minimum weight bipartite matching problem (8) can be solved efficiently. For completeness, we sketch one version of the Hungarian algorithm as its solution [17].

For each edge $(a, b) \in \mathbb{E}$, define the *reduced weight* by $\omega^r_{ab} = \omega_{ab} - p_a - p_b$, where $p := (p_v, v \in \mathbb{A} \cup \mathbb{B})$ is called a price vector indexed by vertices.

Sufficient condition of optimality: If x is a perfect matching in $\mathbb{G}$, and both the following hold:

$$\omega^r_{ab} \geq 0, \ \forall (a, b) \in \mathbb{G} \tag{10a}$$
$$\omega^r_{ab} = 0, \ \forall (a, b) \in x \tag{10b}$$

then x is a minimum weight perfect matching.[5]

See [18] for its proof. The condition suggests two invariants that we will maintain to compute an optimal solution. We initialize a matching $x = \emptyset$ and a price vector p with $p_v = 0$, $v \in \mathbb{A} \cup \mathbb{B}$, which satisfy (10).

Definition 1: Given a matching x and a price vector p, define a path P from $a \in \mathbb{A}$ to $b \in \mathbb{B}$ as a *good* path if

1) both endpoints a and b have not been matched in x;
2) P alternates edges out of x with those in x;
3) every edge in P is tight, i.e., has zero reduced weight.

According to the definition of a good path, it must include an odd number of edges with the first and last ones out of x. The significance of good paths lies in that they enable us to increase the cardinality of x while maintaining both invariants.

Step 1: Matching augmentation:

Given a good path P, $x = x \oplus P$.

Here $\oplus$ denotes the symmetric difference, i.e., $x \oplus P := (x \backslash P) \cup (P \backslash x)$. To see how the matching augmentation works, we can consider it as removing from x the edges that also belong to P and adding in the ones from P that are not in x. Recall that a good path is x-alternating with the first and last edges out of x by definition, thus $|P \backslash x| = |P \cap x| + 1$, i.e., the path augmentation increases the cardinality of x by 1. The two invariants still hold since no reduced weight has changed and all edges in P are tight. After at most N such augmentations, a perfect matching can be attained.

In order to search for a good path efficiently, breadth-first search (BFS) with the enforcement of x-alternation is applied. Given the current matching x, we start a graph search from the first unmatched vertex of $\mathbb{A}$, which is defined as layer 0 of the search tree. Recall the definition of a good path, only tight edges are considered to compose P. Hence we obtain layer 1 of vertices in $\mathbb{B}$ from layer 0 by BFS. Note that if any vertex in layer 1 is unmatched in x, we are done and have an one-edge good path. Otherwise, instead of BFS, in layer 2 we include only vertices matched with those in layer 1 since P has to be x-alternating. Then in any

[4] $\alpha_{ab}, d_{ab}, \tau_{ab}$ and $\alpha_{aj}, d_{aj}, \tau_{aj}$ are used interchangeably when $a \in \mathbb{I} \cup \tilde{\mathbb{I}}, b \in \mathbb{B}_j \cup \mathbb{B}^d_j$.

[5] We abuse x to denote the set of edges that link the matches in x.

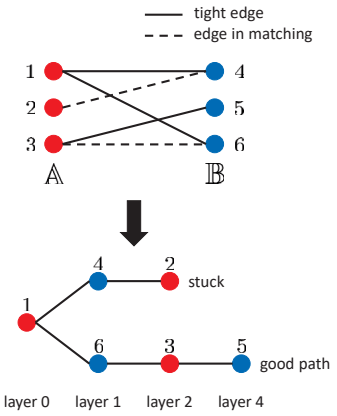

Fig. 2. An illustrative example of searching for a good path.

odd layer we switch back to BFS, and in any even layer we enforce x-alternation, until either we attain a good path in an odd layer or we are stuck in an even layer. See Fig. 2 for an illustrative example.

There is a possibility that the search fails to return a good path of odd length that links two unmatched vertices and gets stuck in an even layer. Under this circumstance, we will maintain the current matching x and update the price vector p in a way that gives rise to a good path. First define $\mathbb{S} \subseteq \mathbb{A}$ as the set of vertices that appear in the even layers of the search tree and $\mathbb{N}(\mathbb{S}) \subseteq \mathbb{B}$ as the set of neighbors of vertices in $\mathbb{S}$ only via tight edges. $\mathbb{N}(\mathbb{S})$ also means the set of vertices that appear in the odd layers of the search tree, since every vertex in an odd layer found by BFS belongs to $\mathbb{N}(\mathbb{S})$ by definition and every vertex in $\mathbb{N}(\mathbb{S})$ is bound to be reached in the search tree, which is only stuck in an even layer when BFS has no tight edge to explore.

***Definition** 2:* $\mathbb{S}$ is defined as a *good* set for two features:

1) $\mathbb{S}$ has an unmatched vertex, i.e. the vertex in layer 0;
2) Each vertex in $\mathbb{N}(\mathbb{S})$ is matched with one in $\mathbb{S}$ by x.

Step 2: Price update:

Given a good set $\mathbb{S}$, as well as $\mathbb{N}(\mathbb{S})$,

$$p_v = \begin{cases} p_v + \Delta, & v \in \mathbb{S} \\ p_v - \Delta, & v \in \mathbb{N}(\mathbb{S}) \end{cases}$$

where Δ is the largest possible value subject to invariants, as explained later.

It can be verified that

1) ω^r_{ab} remains constant for edge (a, b) with $a \notin \mathbb{S}$ and $b \notin \mathbb{N}(\mathbb{S})$;
2) ω^r_{ab} remains constant for edge (a, b) with $a \in \mathbb{S}$ and $b \in \mathbb{N}(\mathbb{S})$;
3) ω^r_{ab} increases by Δ for edge (a, b) with $a \notin \mathbb{S}$ and $b \in \mathbb{N}(\mathbb{S})$;
4) ω^r_{ab} decreases by Δ for edge (a, b) with $a \in \mathbb{S}$ and $b \notin \mathbb{N}(\mathbb{S})$.

Recall that if either endpoint of an edge in x is reached in the search tree, both endpoints are reached. Thus each edge in x can only be categorized to either the first or second case, and its reduced weight sticks to 0, which guarantees (10a). In terms of (10b), the third case has no effect, but the fourth one is potentially hazardous in that the reduced weights of some edges may be negative after going down by Δ. To maintain (10b), Δ is therefore set to the largest possible value that zeros out the reduced weight of an edge belonging to the fourth case while keeping those of others nonnegative. Note that $\Delta > 0$ since in this case none of the edges is tight.

The search tree regrows as a newly tight edge from the price update is added. Suppose edge (a, b) is the newly tight edge, when we search again from the same unmatched vertex, same branches are still explored since edges along them are still tight. Thus the search tree will remain unchanged until vertex a is reached. Then the originally stuck path that ended at vertex a can be now extended, since a new branch linked by edge (a, b) is additionally found by BFS. In this sense, each price update makes progress towards a good path, and such progress required is finite, i.e., at most N price updates for the N vertices in $\mathbb{B}$.

The Hungarian algorithm is summarized in **Algorithm 1**. We just need to augment x N times from $\emptyset$ to a perfect matching, and in each augmentation there are at most N price updates to include all vertices of $\mathbb{B}$ in the search tree. Since each iteration mainly consists of BFS and computing Δ, both of which can be implemented within time of $O(N^2)$, the Hungarian algorithm has a time complexity of $O(N^4)$.

Algorithm 1: Hungarian algorithm

Input: $\mathbb{G} = (\mathbb{A} \cup \mathbb{B}, \mathbb{E})$, and ω;
Output: x;
Initialization: $x \leftarrow \emptyset$, and $p_v \leftarrow 0, v \in \mathbb{A} \cup \mathbb{B}$;
while x is not perfect **do**
  start a search tree with the first unmatched vertex $a \in \mathbb{A}$, and mark layer $k \leftarrow 0$;
  while the tree is not stuck and no other unmatched vertex is found **do**
    if k is even **then**
      attain layer $k + 1$ via tight edges by BFS;
      $k \leftarrow k + 1$;
    else
      attain layer $k + 1$ via matches in x;
      $k \leftarrow k + 1$;
    end if
  end while
  if an unmatched vertex $b \in \mathbb{B}$ is found **then**
    the path P from a to b is good;
    $x \leftarrow x \oplus P$;
  else
    $p_v \leftarrow p_v + \Delta$ for vertices v in the even layers;
    $p_v \leftarrow p_v - \Delta$ for vertices v in the odd layers;
    (the largest possible Δ subject to invariants)
  end if
end while
return x.

IV. Numerical Results

We illustrate with a case study. Suppose currently there are $I = 25$ EVs that require battery swapping and $J = 3$ stations as assignment candidates. Fix $T_m = 6$, i.e., we only look at 6 time slots ahead. Other parameters are randomly generated. For instance, τ_{ij}'s arbitrarily take discrete values between 1 and T_m, then d_{ij}'s are random with an average 5 times the corresponding τ_{ij}. We then let $\alpha_{ij} := 0.02d_{ij} + 0.1\tau_{ij}$. For simplicity, set all EVs' driving ranges

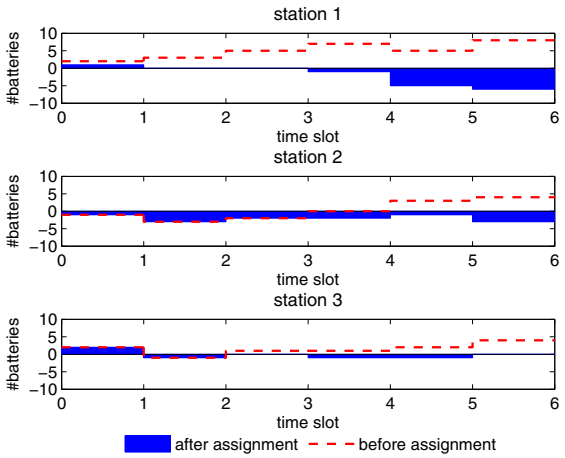

Fig. 3. Optimal assignment of our test case.

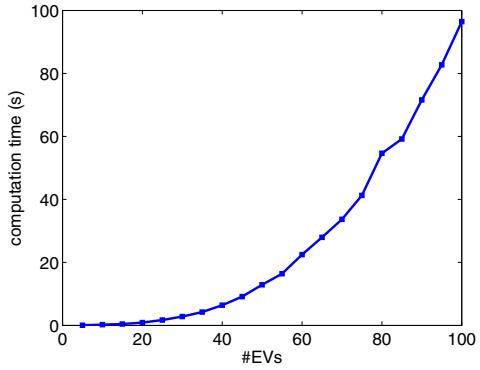

Fig. 4. Scalability (#stations=10).

sufficiently large to reach each station. The parameters of stations, including n_j^0's, $c_j(t)$'s and $\sum_{i\in\mathbb{I}_p} M_{ij}(\tau_{ij}^{-1}(t))$'s, are all randomly picked within certain ranges. Given that, $(\tilde{n}_j(t), t = 1, 2, \ldots, 6, j = 1, 2, 3)$ is attainable, as the red dash lines show in Fig. 3.

The proposed approach efficiently computes an optimal assignment; see Fig. 3 for how the number of available batteries at each station evolves after the assignment. Batteries at station 1 in the first half of the time horizon are almost fully utilized to avoid unduly congesting stations 2 and 3. In the second half all stations run out of batteries. Then (7) provides the optimal assignment. For this test case, the proposed approach achieves a minimal cost of 49.00 with 22.00 total EV cost and 27 total station congestion. In contrast, a heuristic that assigns to each EV its nearest station incurs a cost of 96.46, including 9.46 total EV cost and 87 total station congestion. Hence an optimal assignment achieves a 49.20% improvement for this case.

We check the computational efficiency of the proposed approach by scaling up the number of EVs that require battery swapping while fixing other parameters with the number of stations $J = 10$. The computation time required to run our algorithm on a normal laptop PC (Intel Core i7-3632QM CPU@2.20GHz, 8GB RAM, and 64-bit Windows 10 OS) is shown in Fig. 4.

V. Conclusion

We formulate the problem of optimal station assignment for EV battery swapping that takes into account both temporal and spatial couplings. We show that the problem can be reduced to the minimal weight perfect bipartite matching problem. This leads to an efficient polynomial-time solution. Numerical examples suggest that the proposed approach performs much better than greedy heuristics.

References

[1] C2ES, "Climate TechBook," *Center for Climate and Energy Solutions, US: www.c2es.org/energy/use/transportation*, 2016.

[2] EIA, "Monthly energy review," *Energy Information Administration, US Department of Energy: www.eia.gov/totalenergy/data/monthly/*, 2015.

[3] R. Li, Q. Wu, and S. S. Oren, "Distribution locational marginal pricing for optimal electric vehicle charging management," *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 203–211, 2014.

[4] P. You, Z. Yang, M.-Y. Chow, and Y. Sun, "Optimal cooperative charging strategy for a smart charging station of electric vehicles," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2946–2956, 2016.

[5] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, 2013.

[6] Z. Yang, L. Sun, J. Chen, Q. Yang, X. Chen, and K. Xing, "Profit maximization for plug-in electric taxi with uncertain future electricity prices," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 3058–3068, 2014.

[7] N. Chen, C. W. Tan, and T. Q. Quek, "Electric vehicle charging in smart grid: Optimality and valley-filling algorithms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1073–1083, 2014.

[8] L. Zhang, V. Kekatos, and G. B. Giannakis, "Scalable electric vehicle charging protocols," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1451–1462, 2017.

[9] R. Deng and H. Liang, "Whether to charge an electric vehicle or not? A near-optimal online approach," in *Proc. of IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5, 2016.

[10] P. You, Z. Yang, Y. Zhang, S. H. Low, and Y. Sun, "Optimal charging schedule for a battery switching station serving electric buses," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3473–3483, 2016.

[11] Y. Li, C. Zhan, M. de Jong, and Z. Lukszo, "Business innovation and government regulation for the promotion of electric vehicle use: lessons from Shenzhen, China," *Journal of Cleaner Production*, 2015.

[12] X. Tan, B. Sun, and D. H. Tsang, "Queueing network models for electric vehicle charging station with battery swapping," in *Proc. of IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 1–6, 2014.

[13] B. Sun, X. Tan, and D. H. Tsang, "Optimal charging operation of battery swapping stations with QoS guarantee," in *Proc. of IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 13–18, 2014.

[14] P. You, S. H. Low, W. Tushar, G. Geng, C. Yuen, Z. Yang, and Y. Sun, "Scheduling of EV battery swapping, I: centralized solution," *arXiv preprint arXiv:1611.07943*, 2016.

[15] P. You, S. H. Low, L. Zhang, R. Deng, G. B. Giannakis, Y. Sun, and Z. Yang, "Scheduling of EV battery swapping, II: distributed solutions," *arXiv preprint arXiv:1611.10296*, 2016.

[16] P. You, S. H. Low, Z. Yang, Y. Zhang, and L. Fu, "Real-time recommendation algorithm of battery swapping stations for electric taxis," in *Proc. of IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5, 2016.

[17] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[18] T. Roughgarden, "Minimum-cost bipartite matching," *A second course in algorithms-Lecture 5: http://theory.stanford.edu/ tim/w16/l/l5.pdf*.